C16/MSX

periodico mensile per Commodore 16 e MSX
con listati di giochi e routines

Distribuzione per l'Italia: Messaggerie Periodici S.p.A.
aderente A.D.N., viale Famagosta 75, Milano,
tel. 84.67.545 - Categoria postale gruppo 3

L. 11.000

Dicembre 1990 - n. 41 anno VI

Registraz. n. 557 del 18/10/86 presso il Trib. di Milano
Gruppo Editoriale International Education srl: direz., redaz., amm.ne:
viale Famagosta 75 - Milano
Direttore responsabile: Graham Johnson

GAMES GAMES

12 giochi 12

MSX

C16 compatibile col PLUS 4

sommario

C16/MSX E' IN EDICOLA OGNI VENERDI' DELL'ULTIMA SETTIMANA DEL MESE

Cosa contiene la Cassetta?

C = 16:

1. Tennis
2. U-Boot
3. Quiz
4. Airwolf
5. Footman
6. Plusfile

MSX:

1. Zanac
2. Beret
3. Playball
4. Xevious
5. Arkan
6. Wonder
7. Mistero

Desidero abbonarmi alla rivista 16/MSX allo speciale prezzo di £ 100.000 per 10 copie.

COGNOME ______________ NOME ______________

VIA ______________

CAP ______ CITTÀ ______________ PROV. ______

Allego assegno ☐ ric. versamento sul c/c postale n. 11319209 ☐ intestato a Gruppo Editoriale International Education.

Le richieste di abbonamento andranno inoltrate a:
Gruppo Editoriale International Education
Viale Famagosta, 75 - 20142 Milano - tel. 89502256

Da oggi potrai abbonarti a 16/MSX e ricevere la tua rivista comodamente a casa semplicemente sottoscrivendo uno speciale abbonamento per 10 numeri allo specialissimo prezzo di £ 100.000 anziché £ 110.000.
Potrai così assicurarti la tua copia, risparmiare subito 10.000 lire e avere la sicurezza del prezzo bloccato per tutta la durata dell'abbonamento.

GAMES VIDEO GAMES

AVVERTENZE

Questa cassetta è stata registrata con cura e con i più alti standard di qualità.
Leggete con attenzione le istruzioni per il caricamento. Nel caso in cui, per una ragione qualsiasi, trovaste difficoltà nel caricare i programmi, spedite la cassetta al seguente indirizzo:
Gruppo Editoriale International Education srl
Viale Famagosta, 75
20142 Milano
Testeremo il prodotto e, nel caso, lo sostituiremo con uno nuovo senza costi supplementari.

attenzione! attention! look out! achtung!

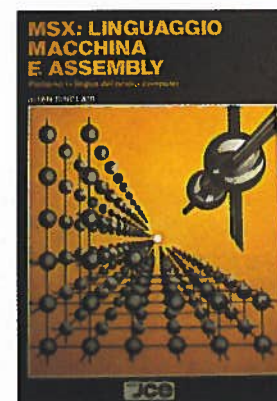
Occhio all'azimut

Per la buona lettura della cassetta occorre che la testina del registratore sia pulita ed allineata col nastro. Se così non fosse potrebbe accadere che sul video appaia "error". Pulite allora la testina del registratore con un cottonfioc imbevuto di alcool. Se nonostante questa operazione il computer continua a non caricare bene prendete un cacciavite ed agite direttamente (in senso orario o antiorario) sulla vite apposita di regolazione dell'azimut.

Se avete un Commodore 16 *digitate LOAD e RETURN, quindi avviate il registratore.*
Per un buon caricamento dei programmi è opportuno tenere il registratore lontano dal monitor e dall'alimentatore.

Se siete i possessori di un MSX *per caricare ogni singolo programma digitate RUN "CAS:" e RETURN.*
Ogni eventuale variazione apparirà in calce alle recensioni dei giochi alla rubrica MSX Challenge.

Recensione libri

MSX: LINGUAGGIO MACCHINA E ASSEMBLY
di Ian SINCLAIR
Edito da: JCE
Pagine: 212
Prezzo: 25.000 lire

In genere, chi usa un computer MSX e sa già scrivere ed utilizzare programmi BASIC, è spesso riluttante ad intraprendere il passo successivo, cioè imparare a servirsi del linguaggio usato dal microprocessore Z80, il cuore dello standard MSX.
Questa rilutttanza è comprensibile. Molti libri che trattano dello Z80 presumono che il lettore conosca già non solo i termini usati, ma anche gran parte del funzionamento del chip del microprocessore. Altri libri, invece, appaiono agli occhi del principiante, come se fossero scritti in una lingua sconosciuta senza sottotitoli. Altri ancora hanno un inizio promettente, ma poi perdono per strada il lettore inesperto, perché affrontano improvvisamente temi assai più difficili o argomenti, come le routines aritmetiche, di ben scarsa utilità per la maggior parte dei lettori.
Questo libro è diretto al vero principiante del linguaggio macchina: chi possiede un computer MSX e sa programmare in BASIC, ma non ha la più pallida idea di quanto avvenga all'interno del suo computer. L'autore di questo libro non si è proposto lo scopo di far diventari i lettori degli esperti nella programmazione in linguaggio macchina dello Z80, perché potrebbero arrivarci solo con molta esperienza, molta pratica, molte letture ed un reale desiderio di risolvere tutti i problemi.
L'autore non pretende neppure di spiegare tutto ciò che lo Z80 è in grado di fare. Ciò che crede di poter ottenere, invece, è di presentare ai lettori l'inizio di un grosso argomento e di fare capire alcuni dei perché e dei percome della programmazione dello Z80.
Un nuovo, immendo mondo si apre davanti agli occhi dei lettori, che potranno capire meglio come funziona il computer e che potranno finalmente comprendere libri più complessi sulla programmazione in linguaggio macchina.
Bisogna però chiarire bene un punto: questo tipo di programmazione non è mai facile. Può diventare familiare, può diventare un normale modo di operare, ma non sarà mai facile. Anche l'imparare è un compito che richiede impegno, un certo sforzo per capire che cosa avviene ed un po' di tempo per fare esercizio sul vostro computer.
In conclusione, questo è davvero un buon libro e per aiutare il lettore, l'autore ha incluso la descrizione di un utilissimo programma: lo ZIN assembler, già pubblicato negli scorsi numeri di questa rivista.

EDITORIALE

La conclusione di un anno e l'inizio del successivo è tempo di festeggiamenti ma anche di consuntivi. Si guarda all'anno appena passato per capire quello che inizia, così vogliamo dare un'occhiata alla situazione attuale dell'universo computer.
Per questo, in questo numero troverete un'articolo dedicato alle tendenze del mercato, mentre ora vorrei soffermarmi su alcune considerazioni di cui è facile rendersi conto.
Il 1990 è un anno da ricordare nella storia dei computer: dopo la prematura scomparsa dello standard Msx, dopo l'estinzione dei computer ZX e con la dipartita degli ultimi Commodore 64 e derivati, quest'anno ha segnato la definitiva scomparsa dal mercato dei computer ad 8 bit.
Piano piano - rispetto ai ritmi informatici - gli home computer che hanno segnato un'epoca sono scomparsi dai negozi, specializzati o meno che siano. Sul mercato ora si trovano solo home computer a 16 bit che si riducono a due standard di base: i PC Compatibili e i 68000 come Commodore Amiga e Atari ST. Sono scomparsi tutti quegli home computer più o meno famosi che spuntavano come funghi nei primi anni 80. E stessa sorte subiranno le console per videogiochi: sul mercato già sono apparse quelle a 16 bit e vedrete che, per esempio, tra un anno esatto si parlerà poco del Sega Mastersystem a 8 bit che sarà soppiantato dal consanguineo Sega Megadrive a 16 bit.
E' forse con un po' di nostalgia che scrivo tutto questo, anche perché sono cresciuto informaticamente grazie al passaggio da un sistema all'altro, partendo dal primo mitico Vic 20 che molti, sono sicuro, ricorderanno con una lacrimuccia negli occhi.
Ma ora basta! Chiudiamo un'epoca e apriamo la nostra rivista. Buon lavoro & Buon divertimento!

LA REDAZIONE

sfida al commodore

1. TENNIS

Pensate che già nel tredicesimo secolo in Francia si giocava un gioco simile al tennis, praticato con le mani nude e utilizzando una palla di pelle di pecora. Nel quindicesimo secolo si cominciarono ad usare le racchette e la rete di divisione e gli inglesi divennero i primi cultori del tennis moderno, che si definì come lo conosciamo adesso nel secolo scorso, attorno al 1875. Il tennis viene giocato da sportivi dei due sessi e il campo può essere fatto su vari tipi di superfici, in erba, terra o materiale sintetico. La larghezza del campo varia a seconda se si gioca in singolo o in doppio e in quest'ultimo caso comprende anche i due corridoi laterali.
Il gioco comincia con un servizio, cioè con la battuta della palla e il servizio passa da un giocatore all'altro, a seconda dei turni di gioco, detti game. Per vincere un game bisogna fare 4 punti, detti convenzionalmente 15, 30, 40 e game. Quando si arriva a 6 games vinti, si vince il set (purché con un vantaggio di almeno due games; altrimenti si continua). Una partita si disputa normalmente al meglio di 3 o 5 set.
Nel gioco che vi presentiamo troverete tutte le regole del vero tennis, in un'avvincente simulazione di una partita di torneo.
Potete giocare da soli o contro un vostro amico: con F1 selezionate il numero dei giocatori, con F2 il livello di difficoltà, con F3 il numero di sets e con F7 la velicità del gioco. Se preferite usare la tastiera, premete Return per selezionarla e usate i tasti:
Z - sinistra X - destra ; - su/giù
Return per tirare.
Per iniziare la partita premete Fire per battere. Quando l'avversario risponde al vostro colpo, portatevi sulla traiettoria della palla, premete Fire per qualche istante per caricare il colpo (la forza è indicata nel riquadro in alto) e quindi premete ancora Fire per tirare.

2. U-BOOT

Durante i primi anni della II Guerra mondiale gli U-Boot tedeschi misero quasi in ginocchio i trasporti via mare alleati, affondando centinaia di navi su tutte le rotte.
Infatti i sottomarini sono tuttora un'arma micidiale, grazie al fatto di essere completamente invisibili in immersione e di poter attaccare tramite i siluri restando qualche metro al di sotto della superficie a quota periscopica.
Come già saprete i sottomarini utilizzano un doppio sistema di propulsione: motori diesel in superficie, più potenti ma che necessitano d'aria per funzionare e motori elettrici, che possono funzionare in immersione ma che però consumano una grande quantità di energia elettrica, esaurendo rapidamente le batterie di accumulatori; quando le batterie sono scariche è quindi necessario riemergere per accendere i motori diesel e ricaricare le batterie tramite appositi generatori. Per provare l'ebbrezza di comandare un sommergibile, caricate il gioco e collegate il

joystick in porta 1,

o usate i tasti:

Rerturn - per scendere

Inst/Del - per risalire

Shift - per lanciare i siluri.

Cercate di calcolare bene la distanza e la velocità del bersaglio e fate attenzione alla carica delle batterie e agli allarmi aerei!

3. QUIZ

Nell'intento di unire l'utile al dilettevole, vi proponiamo un altro gioco appartenente alla serie dei programmi didattici per imparare l'inglese divertendosi.
Dovrete infatti ricordare la posizione di 9 oggetti occultati nelle caselle sullo schermo e trovarli rispondendo alle domande in inglese del computer.
Il programma permette di giocare con tre livelli di difficoltà, ognuno con un diverso tipo di gioco.
Premete A, B o C, seguiti da Return per scegliere il livello e vedrete comparire le nove caselle e gli oggetti di cui dovrete imparare il nome e la posizione.
Vediamo insieme alcune delle frasi utilizzate e il loro significato:

Put the... in a box *metti il... in una casella*
Where is the... *dov'è il...*
Is the... there? *è lì il... ?*
Try again *prova ancora*
Here's the... *ecco qui il...*
You found the... *tu hai trovato il...*
You found these *tu hai trovato questi*
Where is this face? *dov'è questa faccia?*
Here's is the face *ecco qua la faccia*

Per giocare, muovete il riquadro con la barra spazio e scegliete col tasto Return.

4. AIRWOLF

Anno 2100. Base Squawk ZQ23A.
Il tenente Lawson viene improvvisamente svegliato dal trillo del telefono rosso: stamattina lo squillo sembra urlare nelle povere orecchie del tenente.
«Emergenza, signor tenente! E' stata attaccata la piattaforma orbitale 2MV237!!»
Il tenente Lawson sapeva benissimo cosa volevano dire quelle poche concitate parole. Bisognava partire al più presto per la prima linea e cercare di fermare l'orda di extraterrestri che cercavano di penetrare nel nostro sistema difensivo.
La richiesta venne subito accettata dal tenente, che senza perdere un solo attimo era già al posto di comando della sua astronave da battaglia.
Una volta collegato il joystick in porta 1, l'astronave è già pronta alla terribile battaglia. Gli invasori attaccano a ondate successive, ogni volta cambiando il tipo e le caratteristiche delle loro astronavi da battaglia.
Fate attenzione alla temperatura del cannone laser, che non può sparare a lungo ad una elevata cadenza di tiro, pena il surriscaldamento e la temporanea inutilizzabilità. Potete giocare da soli o in due, selezionando uno o due giocatori muovendo lateralmente la leva e premendo Fire.

5. FOOTMAN

Nel corso degli ultimi anni il lavoro di guardiano si è fatto sempre più pericoloso e, paradossalmente, sempre meno remunerativo. Pensate che così non si è più trovato neanche un uomo che facesse la guardia nel recinto dei pulcinoidi che, improvvisamente impazziti, sono diventati un po' pericolosi da tenere a bada.
Si è allora deciso di istituire dei veri e propri turni di controllo, in modo che tutti contribuiscano al controllo dell'allevamento.
Per fare la vostra parte, collegate il joystick in porta 1 e premete Fire per iniziare.
Dovete cercare di evitare il contatto coi pulcinoidi e distruggere tutti i parassiti che si aggirano per il recinto.
Quando siete in difficoltà, potete premere la barra di spazio per avere qualche attimo di tregua; ma potrete usare questa possibilità una volta sola.

6. PLUSFILE

Il Plusfile è un database universale, che permette di creare e gestire un archivio su misura, dall'agenda indirizzi all'elenco dei dischi.
Vediamo subito le istruzioni per l'uso, grandemente facilitato dal menù.
Per creare un nuovo archivioi, premete il tasto M, digitate il numero dei campi, da 2 a 10, quindi date per ogni campo il titolo e la lunghezza. A operazione conclusa, ricompare il menù.
Per introdurre dei dati, premete il tasto A e digitate, campo per campo, i dati da introdurre. Durante l'introduzione potete usare il tasto Inst/Del per cancellare e i tasti cursore verso sinistra o verso l'alto per tornare al campo precedente.
Per cancellare un record, selezionare C e digitare il numero del record desiderato. Per modificarlo, premete E e procedete alla modifica.
Premendo il tasto B si selezionano i campi da visualizzare o stampare, senza modificare i dati in archivio.
Per visualizzare i dati in memoria, premete V; se i record sono stati ordinati, compare anche l'indicazione del campo scelto per l'ordinamento. Per arrestare l'output dei dati e per farlo ripartire premete un tasto qualsiasi, mentre per tornare al menù premete Run/Stop.
I dati in memoria possono essere ordinati, secondo un qualunque campo, in ordine ascendente e discendente. E' anche possibile fare una ricerca tra i vari records, immettendo il nome da ricercare sotto il campo corrispondente.
Per stampare i dati, premere P e quindi attivare l'output premendo V o R.
Per non perdere i dati ad ogni accensione o spegnimento, è necessario salvarli su disco o su nastro. Premete S, quindi C per salvataggio su cassetta e D su disco, poi date il nome del file e infine scrivete un titolo o una data come riferimento futuro.
Per ricaricare i dati, premete L, scegliete la periferica da utilizzare e scrivete il nome del file, che può essere lungo fino a 16 caratteri. Sulla cassetta potrete trovare un file di esempio, chiamato RUBRICA, subito dopo il programma.

LISTATE CON NOI

FUNZIONI GEOMETRICHE

C16

Il listato di questo mese, pur essendo abbastanza breve, ci permette di ottenere ben due funzioni: il disegno di una funzione matematica e la stampa del grafico ottenuto.
Per i nostri lettori più "matematici", la funzione matematica disegnata ha la formula:

af(x) = y–sgn(x)∗sqr(abs(b∗x–c))

ed è possibile variare i 3 coefficienti a, b, c, per ottenere figure diverse.
Per prima cosa, dopo aver digitato il listato e dato il Run, definite il numero dei punti da disegnare; potete partire da 100, per pochi punti, fino a 10000 per una figura molto fitta.
Quindi digitate i 3 parametri, separati da una virgola; alcuni esempi sono 10,200,50 oppure 1,150,200.
Il primo numero influenza la distribuzione dei punti, il secondo la dimensione della figura e il terzo la dispersione dei punti. Infine definite il centro della figura; premendo Return verrà automaticamente tracciata al centro dello schermo.
A disegno terminato, il computer emetterà un suono: premete un tasto per continuare.
A questo punto, se volete l'hardcopy, cioè la stampa su carta, premete S.
Attenzione: a causa della limitata memoria del C16, questa opzione funziona solo sul Plus/4. Per ottenere la stampa anche sul C16, isolare la routine di hardcopy cancellando tutte le linee dalla 100 alla 280 e date il Run.

```
220 GETKEY A$:IF A$="S"THEN CLR:SCNCLR:GOTO 655 35: ELSE END
230 OPEN 1,4 : EN$=CHR$(14):DE$=CHR$(15):PRINT#1 : PRINT#1,EN$
240 EN$=CHR$(14):DE$=CHR$(15):PRINT#1,EN$
250 AP$="NUMERO DEI PUNTI:":PRINT#1,AP$;AZ
260 BA$="PARAMETRI:     A:":BB$="                    B:":BC$="                    C:"
270 PRINT#1,BA$;A:PRINT#1,BB$;B:PRINT#1,BC$;C
280 PRINT#1,DE$:PRINT#1:PRINT#1:CLOSE 1
290 OPEN 4,4 : DIM Z%(199) : S=8192:REM INIZIO BITMAPPING
300 FOR I=39 TO 0 STEP-1:FOR J=0 TO 24
310 FOR K=0 TO 7 : Q = J*8+K
320 Z%(Q) = Z%(Q) + (PEEK(S+320*J+I*8+K))*2↑Y1
330 X$ = X$ + CHR$((Z%(Q)AND127)+128)
340 Z%(Q) = Z%(Q)/128
```

```
100 COLOR 0,1:COLOR4,1:COLOR 1,2 : SCNCLR : PRINT "FUNZIONI GEOMETRICHE"
110 PRINT "DISEGNATE CON LA FORMULA:":PRINT
120 PRINT "A(X) = Y-SGN(X)*SQR(ABS(B*X-C))"
130 DEF FN A(X)=Y-SGN(X)*SQR(ABS(B*X-C))
140 GRAPHIC 0:AZ = 10000:KX = 160:KY = 100
150 PRINT "INPUT DEI PARAMETRI" : INPUT "NUMERO DEI PUNTI                :";AZ
160 INPUT "PARAMETRI A,B,C              :";A,B,C
170 INPUT "CENTRO X-Y (MAX.320,200)   :";KX,KY : GRAPHIC 1,1
180 FOR T=1 TO AZ:DRAW 1,X+KX,Y+KY:XX=FN A(X):Y=A-X:X=XX:NEXT
190 VOL 5:SOUND 1,770,100 : GETKEY A$:GRAPHIC 0
200 PRINT"VUOI L'HARDCOPY (S/N) ?":GETKEY A$:IF A$="S" THEN GOSUB 230
210 PRINT "UN ALTRO GRAFICO (S/N) ?"
350 NEXT K:NEXT J
360 PRINT#4,CHR$(8)X$ : X$ = ""
370 Y1=Y1+1:IF Y1=7 THEN Y1=0:GOTO 400
380 NEXT I
390 R=1
400 FOR L=0 TO 199
410 X$=X$+CHR$((Z%(L)AND127)+128)
420 Z%(L)=Z%(L)/128:NEXT L
430 PRINT#4,CHR$(8)X$:X$=""
440 IF R=0 THEN 380
450 CLOSE4:RETURN
```

MSX challenge

1. ZANAC

Il pianeta Comna è un luogo ameno dove la vita scorre tranquilla come sulla Terra, è ora sotto la minaccia di un attacco da parte degli esseri abitanti il sottospazio di Trixab che vogliono distruggere ogni forma di vita sul pianeta.
Per salvare Comna hai lanciato nello spazio una navetta, un nuovo prototipo da combattimento iperspaziale denominato ZANAC. L'intera galassia è ora in trepida attesa del mortale duello che dovrai affrontare contro i crudeli Trixabiani.
Il tuo obiettivo sarà la distruzione totale delle flotte e degli insediamenti nemici su Comna fino a raggiungere e distruggere la base del potere centrale. Per questo avrai bisogno di tutto il tuo coraggio, di riflessi prontissimi e della massima determinazione in ogni tua azione.
Il computer di bordo della tua nave Zanac è già programmato per condurti alla base di potere centrale ma la rotta passa per le installazioni e postazioni nemiche e così dovrai distruggerle tutte e difenderti dall'attacco degli stormi di caccia stellari nemici.
Distruggendo le postazioni nemiche e gli stormi di navi aliene sullo schermo compariranno dei bonus che se raccolti ti forniranno di potenti armi e nuova energia per continuare la tua missione. Inizialmente potrai sparare in avanti e di lato, a seconda della direzione laterale in cui ti muovi. Nella parte destra dello schermo troverai il punteggio, il record del giorno e tutte le informazioni fornite dal computer di bordo.

COMANDI
Tasti:
[CURSORI] = Movimenti
[SPAZIO] = Inizio gioco / Fuoco
[STOP] = Attiva/Disattiva Pausa
Joystick in porta 1

2. BERET

Sei il protagonista di un'impresa quasi al limite del suicidio dato che dovrai affrontare da solo centinaia di nemici armati fino ai denti. La missione è certamente una delle più pericolose della tua lunga carriera di marines. Sarai infatti impegnato in continui corpo a corpo con i nemici. Per fortuna, di tanto in tanto, c'è la possibilità di sottrarre un'arma ad uno dei nemici così da poterla utilizzare nei momenti di maggior pericolo.
I nemici indossano divise di diversi colori. Una divisa bianca indica che il personaggio è un comandante. La divisa verde significa che il soldato è esperto in armi marziali; le divise marroni sono indossate dai soldati semplici. I soldati con le divise blu sono quelli da evitare perché sono armati e sparano a vista.
Il gioco prevede due livelli di difficoltà. Nel primo livello dovrai utilizzare le rampe e le piattaforme per sfuggire agli attacchi dei nemici. Fa' particolare attenzione agli esperti di karate poiché tenteranno in tutti i modi di eliminarti. Usa le tue doti acrobatiche per scansarli.
Sorveglia attentamente i comandanti: quando ne vedrai comparire uno, cerca di ucciderlo, perché solamente in questo modo potrai entrare in possesso del lanciafiamme.
Alla fine del ponte si trova un temibile avversario, l'artigliere addetto ai mortai. Usa il lanciafiamme per eliminarlo. Se alla fine di tutte queste traversie sarai ancora vivo sentirai il suono di una sirena che annuncia l'arrivo di autocarri carichi di truppe nemiche. Anche in questo caso l'uso del lanciafiamme sarà particolarmente utile. Nel caso in cui l'arma fosse scarica avvicinati agli autocarri, impugna la baionetta e colpisci con decisione. Alla fine del primo livello in prossimità della base missilistica troverai un campo minato. Uomo avvisato... Giunto al secondo livello compariranno nuovi tipi di avversari. Quando incontrerai il comandante (sempre con la divisa bianca) non esitare a colpirlo per rubargli prontamente il "bazooka". In questo livello, oltre alla truppa che ti assalta, dovrai vedertela con paracadutisti, feroci cani doberman e vari ostacoli del genere. L'obiettivo finale è quello di liberare i tuoi compagni prigionieri. Il nostro marines è facilmente manovrabile facendo uso del joystick: può correre, strisciare, saltare e lanciarsi all'assalto con la baionetta o con le diverse armi che troverà sul suo cammino.

COMANDI
Tasti:
[CURSORI] = Movimenti & Salto
[SPAZIO] = Inizio gioco / Usa baionetta
[SHIFT] = Per sparare o fare a pugni
Joystick in porta 1

3. PLAYBALL

Hamburger, Coca-Cola e...Baseball! Non è certo la dieta ideale per uno sportivo, questo no, ma sono elementi inscindibili dello stereoripo di un americano (leggesi "statunitense"). Gli hamburger li conosciamo tutti molto bene, la Coca-Cola è ormai indispensabile alla nostra esistenza, il baseball... stiamo per presentarvelo.
Questo programma è un'ottima simulazione del gioco del baseball e, tra tutti i giochi sportivi, è sicuramente destinato agli appassionati che, nel nostro Paese, sono sempre di più. In Playball potrete cimentarvi in rapide e combattive azioni, in lanci e ricezioni da ripresa televisiva e in corse mozzafiato per occupare le basi.
Le regole sono quelle del Baseball reale, comun- no invertire e invece di ricevere con la mazza è possibile, anzi necessario, alternarsi e lanciare la palla al ricettore che può essere il computer o un'amico umano. Una volta caricato il programma potrete scegliere il tipo di gioco: tre diversi livelli di difficoltà contro il computer oppure una sfida tra due giocatori umani. Dopo questa selezione si devono assegnare squadre e colori ai due giocatori, umani o elettronici che siano. Dopo tutto questo si potrà giocare. Nello schermo di gioco vedrete il campo di gioco in prospezione tridimensionale e sullo sfondo un cartellone che riporta punteggi, strikes e palle Out.

COMANDI
Tasti:
[CURSORI] = Movimenti

4. XEVIOUS

Questo gioco ripropone per gli Msx 1 un'ormai classico gioco arcade, progenitore della folta schiera di shoot'em'up a scorrimento verticale in cui bisogna sparare a tutto ciò che si muove sorvolando un panorama decisamente vario.
Come appena detto, durante il gioco dovrete sorvolare la superficie di un lontano pianeta con lo scopo di sterminare la razza dominante e per raggiungere l'obbiettivo finale che è la superbase, cuore della difesa nemica.
A disposizione del nostro coraggioso eroe tre incursori siderali dotati di laser e cannone neutronico. Il laser può essere sempre usato senza paura di surriscaldarlo, mentre il cannone neutronico deve ricaricarsi per alcuni istanti prima di poter essere riutilizzato. Con il laser si distruggono i mezzi aerei di difesa del nemico, mentre con il cannone si possono colpire e distruggere le basi e le postazioni difensive di superficie del nemico.
Distruggendo le installazioni nucleari nemiche appaiono dei bonus che se raccolti forniranno nuova linfa vitale all'astronave del giocatore e, ovviamente, anche un sacco di punti.
Questi bonus potenziano notevolmente la massa di fuoco del giocatore.

COMANDI
Tasti:
[CURSORI] = Movimenti
[SPAZIO] = Inizio gioco / Laser
[GRAPH] = Cannone neutronico
[STOP] = Attiva/Disattiva Pausa
Joystick in porta 1

5. ARKAN

Ecco un gioco semplicemente meraviglioso e meravigliosamente semplice. Di che si tratta? Della versione per Msx di un arcinoto gioco arcade che per moltissimo tempo ci ha tenuti tenacemente incollati al video, con ampio sperpero delle nostre finanze.
Lo schermo è riempito di mattoncini e tu controlli una paletta che si muove solo orizzontalmente, con la quale dovrai colpire una pallina per scagliarla contro lo strano muro per demolirlo. Nel momento in cui colpirai i mattoncini, se questi sono speciali, ne succederanno di tutti i colori. I mattoncini speciali, una volta colpiti, genereranno delle capsule energetiche che dovrai cercare di recuperare perché dotate di speciali proprietà.
Alcune capsule ti faranno allungare la paletta, altre faranno incollare temporaneamente la pallina alla paletta, altre ancora rimpiccioliranno la mazza, forniranno palline extra, ti daranno una vita bonus o ridurranno il tuo consumo energetico. Ce ne sono poi altre due con proprietà davvero eccezionali: la prima trasformerà la paletta di una nave spaziale con tanto di doppio raggio laser per distruggere il muro, la seconda aprirà un passaggio per farti arrivare al livello successivo.
Ogni tanto faranno la loro comparsa sullo schermo anche degli strani poligoni alieni, apparentemente innocui, che svolazzeranno sul video con lo scopo di mutare la traiettoria della pallina. Incontrerai poi numerose difficolta, che cresceranno proporzionalmente allo schermo raggiunto, ma stai tranquillo: in questo gioco è tutto così assurdo e irresistibile che la smania del gioco ti attanaglierà in brevissimo tempo.

COMANDI
Tasti:
[CURSORI] = Movimenti paletta
[SPAZIO] = Inizio gioco / Lancia pallina & Spara
Joystick in porta 1

6. WONDER

E' tempo di vacanze natalizie e Ciccio decide di andare in Kenia dove, in un caldo e assolato pomeriggio di Dicembre (?), decide di mollare tutto e di addentrarsi all'interno della foresta tropicale alla ricerca della vita selvaggia. Qualcuno però gli ha suggerito di essere prudente e portarsi con se qualche arma, ma Ciccio, che sente il richiamo della foresta sempre più forte, non vuole saperne di diavolerie moderne e decide di portare con se un primitivo boomerang.
Appena addentratosi nella foresta, il nostro eroe si rende conto che rimanere in vita sarà tutt'altro che facile e rimpiange la civiltà che ora vuole riabbracciare. Ma la strada verso la città sarà dura e gli ostacoli che si troverà davanti saranno così di dinosauro che incontrerà lungo la strada.
Alcune lo renderanno invulnerabile al punto tale che sarà in grado di superare incolume ostacoli notevoli, quali velenosissimi serpenti.
Altre sorprese gli regaleranno uno stupendo skateboard che non si capisce così ci faccia lì. Attento però! La vulnerabilità durerà poco e potrai perdere lo skateboard molto facilmente.
Un consiglio: dai modo a Ciccio di recuperare le forze raccogliendo la frutta sparsa per il sentiero.

COMANDI
Tasti:

MSX challenge

7. MISTERO

MISTERO

Ecco una nuova classica avventura di testo. Per poter utilizzare l'avventura non dovete usare il solito comando LOAD"cas:",R oppure RUN-"cas:", ma dovete usare l'istruzione CLOAD. Quando il caricamento è terminato dovete digitare RUN e premere il tasto di invio.

COME GIOCARE
Giocare una adventure significa guidare il protagonista di una storia verso uno scopo ben preciso, impartendo comandi tipo "PRENDI IL MARTELLO" o "ROMPI IL VASO" formate essenzialmente da un verbo seguito da un sostantivo e dalla pressione del tasto di immissione [RETURN]. Le istruzioni vanno impartite per esteso, in lingua italiana e utilizzando la seconda persona singolare. Potrete spostarvi nelle quattro direzioni cardinali NORD, SUD, EST, OVEST e ordinare SALI e SCENDI o, se la situazione lo richiede, ATTRAVERSA, ENTRA o ESCI. La descrizione del luogo in cui vi trovate e degli oggetti o dei personaggi presenti potrà scomparire a causa dello spostamento verso l'alto (SCROLL verticale) del testo : per rivederla basterà digitare DESCRIVI o GUARDA. Gli oggetti potranno essere manipolati per mezzo dei verbi PRENDI, POSA e INDOSSA o TOGLI (questi ultimi due se si tratta di ornamenti o di vestiti). Per inventariare gli oggetti trasportati dovrete digitare LIST oppure INVENTARIO. L'esame di oggetti, luoghi o personaggi attraverso ESAMINA potrà condurre in molti casi alla scoperta di altri oggetti o di particolari nascosti : se ciò dovesse accadere, lo schermo verrà cancellato e la descrizione del luogo aggiornata. Infine, ricordandovi che la soluzione di una avventura può richiedere anche giorni o settimane di "duro impegno", tenete presente che è possibile registrare su un'altra cassetta disco la situazione corrente di gioco col comando SAVE e ripristinarla in qualsiasi momento col comando LOAD. In seguito a tali istruzioni vi verrà chiesto di inserire la cassetta di destinazione dei dati e premere un tasto, dopodiché lo schermo si cancellerà per il tempo occorrente alla lettura o alla scrittura dei dati. Questo è il vocabolario essenziale di ogni avventura che si rispetti, ma il Dizionario di Gioco (PARSER), contiene molti altri vocaboli, relativi alle situazioni che incontrerete, per cui impartite normalmente le vostre istruzioni, cercando di cambiare i vocaboli nel caso in cui il computer non vi comprendesse. Ora però leggete attentamente il seguente racconto, prassi assolutamente indispensabile per poter affrontare e risolvere l'avventura.

PROLOGO
Lavorare come fotoreporter è già di per sè un'attività entusiasmante ma per i collaboratori del "Today's Mysteries" ogni servizio si trasformava puntualmente in un vero e proprio viaggio nell'ignoto.
Roy Norton era stato assunto da più di un anno nella redazione del giornale, ma la sua occupazione principale fino ad allora era stata quella di scrivere trafiletti per riempire gli spazi vuoti lasciati dai suoi più fortunati - e meglio pagati - colleghi. Un lavoro da "certosino", che lo costringeva a passare ore intere nei locali più bui e polverosi della redazione, scervellandosi per inserire notizie abbastanza interessanti fra le pagine del "Today's Mysteries", senza che questo contribuisse minimamente a migliorare la sua posizione di giornalista.
Ma per ogni reporter arriva, prima o poi, lo scoop della sua vita, il colpo giornalistico che può renderlo famoso: basta tenere gli occhi bene aperti, tutto qui. E Roy li aveva spalancati, gli occhi: così quando sentì parlare del mistero di Hal Parson, aprì bene anche le orecchie!
Scoprì che quello era il nome di uno scrittore di romanzi gialli e che della sua vita privata si sapeva poco o niente. Correvano voci contraddittorie, stranissime, che avrebbero incuriosito chiunque: ma nessuno al giornale sembrava aver voglia di occcuparsene.
Roy pensò che riuscire ad avere un'intervista esclusiva da quel fantomatico maestro dell'Horror e del brivido gli avrebbe procurato la promozione a cui aspirava da tempo. Al diavolo i suoi colleghi! Le loro strane superstizioni e le storie sui giornalisti scomparsi nella villa di Hal Parson... le loro chiacchiere davano all'intera faccenda l'aria di una bidonata pubblicitaria, e a lui non andava che fosse considerato tale il primo vero servizio che era riuscito a farsi affidare...
Ma mentre guidava la sua vecchia Ford rossa verso il parcheggio della villa dove lo scrittore viveva e lavorava, Roy fu assalito da un dubbio atroce: e se i suoi colleghi avessero deciso di organizzargli un orribile scherzo? Si era parlato fin troppo, in sua presenza, dell'impossibilità assoluta di farsi concedere un'intervista da Mr.Parson, del fatto che nessuno conoscesse il suo volto o la sua voce, e di strane storie sulla sua villa fuori città...
Una faccenda davvero poco chiara, pensava Roy salendo i gradini che lo portavano all'ingresso principale di Villa Parson, e come d'istinto toccò il vecchio amuleto che da anni portava appeso al collo: una zanna di lupo donatagli da una vecchia cartomante che aveva intervistato. Gli sembrò, in quell'attimo, di rivivere la scena e si ricordò improvvisamente delle strane parole pronunciate dalla vecchia mentre gli donava l'amuleto: "Un giorno questa zanna di lupo ti salverà, figliolo. Ricordati che se sarai in pericolo ti basterà gridare 'LICANTROPUS' e il potere della zanna ti salverà...ricorda!"
Fu come rivedere un pezzo di film al rallentatore, e le parole della cartomante, in quel momento, non gli sembrarono più tanto bizzarre come la prima volta... Con un brivido si scosse e realizzò di essere giunto in cima alle scale, davanti al portone della villa: avrebbe quasi voluto tornare indietro, ma immaginava lo scherno dei colleghi...
Fu così che Roy Norton si decise, lasciando dietro se la realtà di ogni giorno per incontrare, oltre il portone di Villa Parso, le mille facce dell'ignoto, in un'avventura tra il magico e l'irreale che avrebbe segnato il suo battesimo del fuoco come 'Reporter del Mistero'.

BASIC (Parte 6)

VARIABILI STRINGA E CODICE ASCII

Una variabile numerica, come ben sapiamo, può contenere un valore numerico. Una VARIABILE STRINGA (riconoscibile perché il nome termina con $ dollaro) contiene invece un certo numero di caratteri (anche nessuno, nel caso della stringa nulla) generalmente in codice ASCII.
Ad ogni carattere, perciò, è associato un numero: il valore del codice ASCII corrispondente.
Tuttavia, numeri e stringhe sono due cose distinte. I numeri possono essere usati in espressioni, mentre le stringhe sono "ripetute" dal calcolatore tali e quali. Su di esse è possibile operare soltanto per mezzo di apposite istruzioni, che non abbiamo ancora trattato. Ecco alcuni esempi di errori:

```
A="3"   ("3" è un carattere, non un numero)
C$=12   (12 è un numero, non un carattere)
IF A=C$ THEN ..... (numeri e stringhe non sono confrontabili)
```

A volte, però, può essere utile conoscere il codice ASCII di un carattere o, viceversa, produrre un carattere dato il suo codice (ad esempio, per produrre i caratteri di controllo necessari ad alcune unità video). Esistono, a questo scopo, due funzioni BASIC, denominate ASC (CODE in alcune macchine) e CHR$. Cominciamo dalla prima.
Ecco un programma che chiede un carattere e stampa il valore del codice ASCII corrispondente:

```
10 INPUT "DAMMI UNA LETTERA: ";A$
20 B=ASC(A$)
30 PRINT "IL CODICE DI ";A$;" E' ";B
40 GOTO 10
```

Alla linea 20 compare la funzione ASC, che calcola il codice ASCII del suo argomento, cioé del carattere contenuto nella stringa A$. Se la stringa A$ è lunga più di un carattere, ad esempio "ROMA" o "NO", la funzione ASC calcola il codice del primo carattere (cioé la "R" di "ROMA" o la "N" di "NO").
Due osservazioni. Per prima cosa, abbiamo detto che ASC è una FUNZIONE. Una funzione è un tipo particolare di operatore che, in base ai suoi ARGOMENTI (in questo caso la stringa A$), ritorna un valore. In altre parole, l'intera funzione ASC(A$) si comporta come un valore numerico e come tale può essere impiegata, ad esempio all'interno di espressioni come 3+ASC(A$). In secondo luogo, in BASIC il nome di una funzione indica anche il TIPO del risultato: se il nome termina con $ (dollaro), la funzione RESTITUISCE (equivale a) una stringa, altrimenti (è il nostro caso) restituisce un valore numerico. Proviamo a far girare il programma:

```
RUN

DAMMI UNA LETTERA: A
IL CODICE DI A E' 65
DAMMI UNA LETTERA: C
IL CODICE DI G E' 71
DAMMI UNA LETTERA: M
IL CODICE DI M E' 77
```

e così via fino al [CTRL]-[C] (se il programma sta eseguendo una INPUT, occorre premere [RETURN] dopo il [CTRL]-[C]).
Per evitare di dover terminare un programma con un'interruzione forzata (come [CTRL]-[C]) è preferibile inserire nel programma stesso una condizione che ne determini l'arresto quando lo si desidera.
Possiamo stabilire ad esempio, che il programma esegua un'istruzione END (e quindi si fermi) quando si risponda all'INPUT con un certo carattere. Aggiungiamo queste due istruzioni:

```
25 IF A$ = "*" GOTO 50
50 END
```

Se, in risposta alla INPUT della linea 10, viene dato il carattere # (DIESIS o CANCELLETTO o POUND), la condizione posta nella IF diventa vera ed il programma salta alla linea 50, dove termina.
Data un funzione, se ne esiste un'altra che compie il lavoro opposto (o RECIPROCO), quest'ultima prende il nome di FUNZIONE INVERSA. Tornando alla nostra ASC, esiste una funzione inversa che si chiama CHR$.
La CHR$ produce (restituisce) il carattere avente come codice ASCII il numero dato. Per esempio CHR$(65) corrisponde alla lettera A, CHR$(66) alla B, e così via.
Poiché la funzione CHR$ costituisce un carattere (cioé una stringa di un solo carattere) e non un numero, il nome della funzione termina con $.
Per verificare che CHR$ è la funzione inversa ASC, basta scrivere:

```
PRINT ASC (CHR$ (70))      -› 70
PRINT CHR$ (ASC ("F"))     -› F
```

(lo spazio tra il nome della funzione e la parentesi non è obbligatorio, lo mettiamo solo per meglio evidenziare il nome stesso).
Vediamo ora due programmi che potete usare per familiarizzarvi con le funzioni ASC e CHR$: Segretissimo 1 e Segretissimo 2. Il primo traduce una frase in "codice segreto" (una lettera alla volta), con l'ausilio della funzione ASC. Il secondo interpreta il "codice segreto" e lo riporta in chiaro, mediante la funzione CHR$.
Ecco il primo programma.

```
10 REM --- SEGRETISSIMO 1 ---
20 PRINT "DAMMI LA CHIAVE SEGRETA (1-37)";
30 INPUT K:IF (K‹1) OR (K›37) GOTO 20:REM CONTROLLO
CHIAVE
40 PRINT "SCRIVI UNA LETTERA DA CIFRARE (A-Z, #
PER FINIRE) ";
50 INPUT A$:IF (A$‹"A") OR (A$›"Z") GOTO 40: REM CON-
TROLLO
60 IF A$="#" THEN END
70 B=ASC(A$)+K
```

```
80 PRINT B
90 GOTO 50
```

Qualche commento sul programma.

Le linee 20-30 chiedono la "chiave segreta", che viene assegnata alla variabile K. La linea 30, in particolare, mostra come si costruisce un INPUT CONTROLLATO, che accetti soltanto i valori voluti (se la chiave non è tra 1 e 37, ripete la domanda alla linea 20).
Le linee 40-50, dove il programma chiede una lettera da cifrare, sono un altro esempio di input controllato. Il controllo alla linea 50, però, è diverso dai soliti: confronta stringhe invece di numeri. Un'espressione logica tipo IF A$‹"A" è vera se il primo operando (il contenuto di $) precede, in ordine alfabetico (in realtà, in ordine di codici ASCII, ma fa lo stesso), il secondo operando ("A"). Nei computer che non accettano questo genere di operazione, occorre confrontare i codici ASCII; in questo caso, la linea 50 andrebbe riscritta:

```
50 INPUT A$:IF (ASC(A$)‹ASC("A")) OR
(ASC(A$)›ASC("Z")) GOTO 40
```

La linea 60 consente di terminare il lavoro di cifratura: quando si scrive # viene eseguito l'END ed il programma termina. Se il calcolatore non accetta un END in mezzo al programma, occorre scrivere:

```
60 IF A$="#" GOTO 100

100 END
```

Alla linea 70, finalmente, la lettera viene crittografata, sommando K al valore del suo codice ASCII. La linea 80 stampa il "cifrato segreto" da trasmettere al "comando operativo". Per esempio se K = 4, si ottiene una conversione di questo tipo:

```
A   E   R   E   O
69  73  86  73  79
```

Si vede che ogni valore numerico è ottenuto sommando al codice ASCII la chiave K=4(A=65+4=69). Con Segretissimo 2, il messaggio segreto viene decrittografato. Soltando il "comando operativo", che conosce la chiave può effettuare la decodifica.
Ecco Segretissimo 2:

```
10 REM --- SEGRETISSIMO 2 ---
20 PRINT "DAMMI LA CHIAVE SEGRETA (1-37)";
30 INPUT K:IF (K‹1) OR (K›37) GOTO 20:REM CONTROLLO
CHIAVE
40 PRINT "SCRIVI UN CODICE DA DECIFRARE (0 PER FI-
NIRE)"; 50 INPUT C
60 IF C=0 THEN END
70 B$=CHR$(C-K)
80 PRINT B$
90 GOTO 40
```

Nelle linee 20-30 viene chiesta, come al solito, la chiave segreta. Le linee 40-50 chiedono il codice da decifrare. La linea 60 termina il lavoro se si è battuto 0.
La linea 70 ricostruisce il carattere originario decrittografando il codice: prima viene sottratto K (le operazioni tra parentesi hanno sempre la precedenza), poi viene assegnato a B$ il carattere avente come codice ASCII il valore calcolato, che non è altro se non il carattere originariamente cifrato con Segretissimo 1.
Questa coppia di programmi, come tutti quelli che presentiamo per illustrare le istruzioni BASIC, ha molte manchevolezze.
Per dirne una, in Segretissimo 2 non c'è alcun controllo nell'input di C, col risultato che il programma può "piantarsi" (terminare con errore) alla linea 60 se C-K non corrisponde ad un codice ASCII valido.
D'altronde, per ora ci interessano la chiarezza e la semplicità degli esempi e non vogliamo appesantirli con troppe rifiniture o confondere inutilmente le idee.

UTILIZZO DELLA FUNZIONE CHRS

Un tipico campo di impiego della funzione CHR$ è la generazione dei caratteri SPECIALI, o CARATTERI DI CONTROLLO, richiesti da alcune periferiche per compiere determinate funzioni (ad esempio la cancellazione del video). A volte è possibile inserire questi caratteri tra virgolette, in una normale stringa, ma di solito non risultano poi rileggibili (o appaiono come strani simboli grafici, come nei Commodore), pregiudicando la chiarezza del programma. Conviene quindi utilizzare la funzione CHR$ per produrre questi caratteri.
Supponiamo, ad esempio, che il carattere con codice ASCII 10 abbia come effetto la cancellazione dello schermo, se stampato con una normale PRINT.
Se desideriamo cancellare lo schermo, basta fare:

```
PRINT CHR$(10);
```

o, meglio ancora, mettere il carattere speciale in una variabile stringa dal nome facilmente ricordabile (CS$ per "cancella schermo") ed usare poi questa nel corso del programma:

```
10 CS$=CHR$(10):REM CANCELLA SCHERMO

140 PRINT CS$;

320 PRINT CS$;
```

Altri caratteri speciali permettono di controllare il movimento del cursore sul video, lo stile usato dalla stampante, ecc.
La funzione CHR$ è anche utile per produrre un carattere che, pur essendo stampabile, non può essere inserito nel testo BASIC: le virgolette (").
Supponiamo di voler stampare una frase contenente le virgolette:

```
10 PRINT "STAMPARE LE VIRGOLETTE (") IN BASIC"
```


Il calcolatore accetterebbe il testo compreso tra le prime due coppie di virgolette come la stringa "STAMPARE LE VIRGOLETTE (" e poi segnalerebbe un errore per il resto della frase, considerata fuori dalle virgolette.
Per ovviare a questo inconveniente, alcuni calcolatori (pochi) permettono di introdurre le virgolette battendole due volte. Nel nostro caso dovremmo scrivere:

```
10 PRINT "STAMPARE LE VIRGOLETTE ("") IN BASIC"
```

Con CHR$ è possibile invece una soluzione valida per qualunque dialetto BASIC. Dal codice ASCII risulta che le virgolette hanno codice 34. Possiamo allora scrivere:

```
10 PRINT "STAMPARE LE VIRGOLETTE (";CHR$(34);") IN
BASIC"

Provando ad eseguire, otteniamo:

RUN

STAMPARE LE VIRGOLETTE (") IN BASIC
```

Se le virgolette devono essere usate spesso è meglio assegnarle ad una apposita variabile:

```
10 VI$=CHR$(34):REM VIRGOLETTE

80 PRINT "STAMPARE LE VIRGOLETTE (";VI$;")
IN BASIC "
```

DATI NEL PROGRAMMA

Le istruzioni DATA, READ e RESTORE consentono di inserire nel programma liste di dati (costanti), che vengono poi rilette in modo analogo ad un INPUT dall'esterno. E' possibile, ad esempio, avere un'agenda telefonica scritta nel programma stesso, senza dover inserire ogni volta i dati.

ISTRUZIONI DATA, READ e RESTORE
Spesso è utile INIZIALIZZARE delle variabili, cioè assegnare ad esse dei valori definiti.
Questo può essere agevolmente ottenuto con sequenze di istruzioni del tipo:

```
10 A = 230
20 B = 40
30 C = 155
40 D = 12
50 E = 130
60 F = 18
70 G = 2
80 H = 760
```

L'istruzione DATA consente invece di riunire tutti i dati in un unico punto del programma, di solito all'inizio o alla fine, dove è più leggibile:

```
10 DATA 230, 40, 155, 12, 130, 18, 2, 760
```

L'istruzione READ (leggi) è equivalente alla INPUT, con la differenza che i dati non vengono letti dalla tastiera, ma dalle linee di DATA. Si può quindi scrivere:

```
120 READ A, B, C, D, E, F, G, H
```

e gli otto valori vengono assegnati alle rispettive variabili.
Se vi sono più linee di DATA, anche in punti diversi del programma, sono lette di seguito come se si trattasse di un'unica linea. Con le poche nozioni di BASIC finora viste, queste istruzioni non si rivelano di grande utilità. Possiamo comunque fare un semplice esempio della loro utilizzazione: un'agenda telefonica.
Poiché non abbiamo ancora parlato di come registrare e rileggere dati da nastro o disco, possiamo tenere i dati nel programma stesso, che sappiamo come memorizzare (SAVE) e rileggere (LOAD). Vediamo il programma.

```
10 REM --- AGENDA TELEFONICA --
30 INPUT "NOME ? ";NO$:IF NO$="" THEN END
40 RESTORE
50 READ A$, TE$
60 IF A$="FINE AGENDA" THEN PRINT "NON C'E'.":GO-
TO 30
70 IF A$‹›NO$ GOTO 50
80 PRINT "IL TELEFONO DI ";NO$;" E': ";TE$:GOTO 30
```

```
1000 REM AGENDA
1010 DATA "BIANCHI", "680368"
1020 DATA "ACI", "06-4212"
1030 DATA "BORSA", "6292"
1040 DATA "AIUTO", "113"
9000 REM FINE AGENDA
9010 DATA "FINE AGENDA", ""
```

Per prima cosa, notiamo che i nominativi ed i numeri di telefono vanno introdotti sotto forma di stringhe contenute in linee di DATA (si può proseguire a volontà dopo la 1040), e quindi vengono salvati insieme al programma stesso.
Al RUN il programma inizia chiedendo (linea 30) il nominativo di cui cerchiamo il numero, assegnandolo alla variabile NO$.
A questo punto, la stringa NO$ viene confrontata con la stringa nulla (""): se ci si limita a premere il tasto di invio, il programma esegue un END e termina.
La linea 40 esegue un RESTORE, che riporta all'inizio la lettura dei DATA. Nel nostro caso, alla linea 1010, dove si trova il primo DATA. Il RESTORE è necessario perché l'agenda va letta dall'inizio.
La linea 50 legge due stringhe dai primi due DATA ("BIANCHI" e "680368"), rispettivamente nelle variabili A$ e TE$.
Dopo questa operazione, il PUNTATORE dei DATA si è spostato, cioè la prossima READ leggerà dalla costante DATA successiva.
La linea 60 verifica se, a forza di leggere DATA, si è arrivati all'ultimo ("FINE AGENDA", linea 9010).

BASIC (Parte 6)

In questo caso avverte che il nome non è in agenda e ricomincia (linea 30). La linea 9010 contiene anche una stringa nulla (""), per evitare errori nella lettura dell'ultimo TE$ alla linea 50.
La linea 70 confronta il nome cercato con il primo DATA letto; se non coincidono, torna alla linea 50, dove viene letto il prossimo.
Alla linea 80 si giunge solo se il nome coincideva: viene dunque stampato il numero di telefono.
I DATA sono due per riga solo per leggibilità: avremmo, infatti, potuto metterli anche in una sola riga. I numeri di telefono sono sotto forma di stringhe per consentire l'impiego di caratteri non numerici o zeri iniziali (vedi linea 1020).
Per quanto questo programma sia rudimentale, può venire utile ai possessori di un computer tascabile in BASIC con memoria permanente.

CICLI

ISTRUZIONE FOR...NEXT
Capita spesso di voler eseguire più volte un'istruzione od un gruppo di istruzioni. Supponiamo di voler stampare i primi dieci numeri interi:

```
10 I = 1
20 PRINT I
30 I = I + 1
40 IF I<11 GOTO 20
```

Abbiamo usato I come CONTATORE, per contare cioè quante volte il programma esegue il CICLO (LOOP) costituito dalle linee 20, 30 e 40. Alla linea 30 il contatore viene INCREMENTATO (aumentato di uno) ed alla 40 si controlla se il ciclo è stato ripetuto per il numero voluto di volte.
Tutto questo può essere ottenuto molto più semplicemente:

```
10 FOR I = 1 TO 10
20 PRINT I
30 NEXT I
```

L'istruzione alla linea 10 dice: metti I uguale a 1, esegui le istruzioni che seguono fino al NEXT I, incrementa I di 1, se non ha superato il 10 ripeti, se no continua con l'istruzione che segue il NEXT I. Cioè il nostro programma può anche essere letto nel seguente modo:

```
10 PER (FOR) I che va da 1 FINO A (TO) 10
20 Esegui la linea 20
30 Passa al PROSSIMO (NEXT) I
```

in cui I si chiama VARIABILE DI CONTROLLO DEL CICLO, 1 è il VALORE INIZIALE e 10 è il VALORE FINALE o DI TEST.
Al posto della linea 20 può stare qualunque altra linea, o insieme di linee, che vogliamo sia ripetuto 10 volte.
L'incremento, o PASSO, della variabile di controllo del ciclo è 1 se non specificato, ma può essere indicato esplicitamente con STEP (PASSO, appunto).
Ad esempio, vediamo un programma che calcola i quadrati dei numeri pari compresi tra 8 e 46:

```
10 FOR N = 8 TO 46 STEP 2
20 PRINT "NUMERO: ";N;" QUADRATO: ";N*N
30 NEXT N
```

Partendo da 8 ed aumentando ogni volta di 2, fino a 46, vengono stampati i numeri ed il rispettivo quadrato.
Si può anche contare all'indietro:

```
10 REM --- CONTO ALLA ROVESCIA ---
20 FOR CD = 10 TO 1 STEP -1
30 PRINT CD; " ";
40 FOR I = 1 TO 2000:NEXT I
50 NEXT CD
60 PRINT "0: LANCIATO!"
```

Questa volta abbiamo introdotto un secondo ciclo dentro il primo (linea 40), che ha solo la funzione di far passare un pò di tempo. Si dice che ci sono due CICLI NIDIFICATI (NESTED).
In casi come questo, bisogna ricordare che i cicli vanno chiusi (NEXT) nell'ordine inverso a quello in cui sono stati aperti (FOR). In altre parole, l'ultimo aperto va chiuso per primo e il primo per ultimo.
Un ultimo esempio: un programma che produce la tavola pitagorica della moltiplicazione:

```
10 REM --- TABELLINA PITAGORICA ---
20 FOR I = 1 TO 10
30 FOR J = 1 TO 10
40 PRINT I * J; " ";
50 NEXT J
60 PRINT
70 NEXT I
```

Le istruzioni 30, 40 e 50 costituiscono il ciclo interno che calcola e stampa una riga della tavola, con I costante e J che varia da 1 a 10.
Notate che il PRINT alla linea 40 termina con punto e virgola (;) per non andare a capo durante la scrittura di una riga.
Per ciascuna riga si ripete lo stesso ciclo con un valore diverso di I. Il ciclo esterno (quello in cui varia I) fa ripetere dieci volte il ciclo interno (in cui varia J). La PRINT alla linea 60 serve per andare a capo al termine di ogni linea.
Provate a far girare il programma.
Fatto? Perché la tabellina non è venuta esattamente allineata?
Vi lasciamo trovare la risposta, con un aiuto: l'istruzione

```
45 IF I*J < 10 THEN PRINT " ";
```

mette tutto a posto.

(continua)

DENTRO L'MSX (Parte IVª)

Nello scorso numero avevamo concluso parlando del colore nelle diverse modalità di testo e dell'uso della VPOKE per cambiare la tabella dei colori nella Ram, istruzione usata anche per ridefinire il set di caratteri.
In entrambe le modalità testo, la posizione del cursore viene determinata tramite l'istruzione LOCATE X,Y. Nel modo a 40 colonne è possibile visualizzare fino a 40 caratteri su una linea. Nonostante siano accettati posizioni di colonna fino a 255, solo quelle comprese nell'intervallo 0-39 hanno effetto. Per la modalità a 32 colonne l'intervallo accettabile è da 0 a 31. Un'alternativa all'istruzione LOCATE 0,0 è la stampa su schermo di CHR$(11). Il comando LOCATE controlla inoltre la visualizzazione del cursore:

```
LOCATE 2,4,0 fa si che esso non venga visualizzato
LOCATE 2,4,1 abilita la visualizzazione del cursore
```

Il comando PRINT, che può essere abbreviato con "?", usa i caratteri standard per la formattazione:

1 - Il punto e virgola implica che il carattere di "ritorno carrello" (CR o Carriage Return, CHR$(13)) non venga stampato dopo l'ultima entità da visualizzare. In questo modo la prossima visualizzazione non partirà dalla prima colonna della linea successiva.

2 - Una virgola implica che la prossima entità visualizzata sia allineata con il prossimo carattere (o zona) di tabulazione presente sulla linea. Ogni zona di tabulazione è ampia 14 colonne.

3 - Il segno "+" serve per concatenare delle stringhe.
Per esempio:

```
A$ = B$ + "XYZ";

TAB(n) e SPC(n) devono seguire l'istruzione di PRINT cioè:

PRINT SPC(4);"NOME
```

Si noti che il delimitatore finale " può essere assente.
La variante PRINT USING consente di stampare tabelle numeriche o alfanumeriche in un dato formato.
I quattro principali caratteri di controllo sono:

1 - Il punto esclamativo (!): implica che venga stampato solo il primo carattere di una stringa, come PRINT USING "!";"NOME" che darebbe come risultato N

2 - La barra (/): viene usata nella forma "//" per specificare il numero di caratteri della stringa che devono essere stampati. Il numero dei caratteri stampati è infatti due più il numero di spazi tra le due barre, come, ad esempio, PRINT USING "//";"NOME" che darebbe come risultato NOM. Se la lunghezza del campo supera quella della stringa, vengono aggiunti degli spazi.

3 - Il segno di E commerciale (&): viene usato per inserire una seconda stringa in una prima a partire dalla posizione in cui è posta la &; quindi

```
Q$="NOME":? USING "IN &";Q$ darebbe come risultato IN NOME.
```

Si noti che nel caso venissero fornite più sottostringhe, la sequenza mostrata in precedenza verrebbe ripetuta per ciascuna di esse. Per esempio:

```
Q$="A":Z$="B":? USING "SU $";Q$;Z$ darebbe come risultato SU ASU B.
```

4 - Il segno di diesis (#): consente di stampare i numeri fissando il numero di cifre prima e dopo la virgola decimale. Se il valore è più corto, vengono aggiunti degli 0. Per esempio:

```
? USING "####.####";2,4.684 darebbe 2.004.68.
```

Se viceversa il campo è troppo corto per contenere il dato, allora viene stampato il segno % davanti al valore, oppure viene arrotondato. Per esempio:

```
? USING"##";22444 darebbe come risultato %22444
```

mentre

```
? USING"##.##";22.1234 darebbe 22.12.
```

Se il campo è troppo grande il valore viene giustificato a destra o vengono aggiunti degli zeri.
Per esempio:

```
? USING"##.##";22.2 darebbe come risultato 22.20
```

mentre con

```
? USING "#####";22 si otterrebbe 22.
```

Il numero non può essere più lungo di 24 cifre, nel qual caso verrebbe segnalata una situazione di errore.
Altri caratteri di controllo per i numeri, usati insieme al segno "#", sono: **, $$, la virgola, +, - e ^^^^ (detto "carat").
Un segno più (+) alla destra o alla sinistra del simbolo diesis (#) implica che venga stampato il segno del valore numerico, rispettivamente alla destra o alla sinistra del valore stesso.
In modo analogo, un segno negativo dopo l'ultimo di una

serie di "#" provoca la stampa del segno meno dopo il valore, se questo è negativo.
Quindi:

```
? USING "##-";-2 stamperà 2-
```

La combinazione doppio asterisco (**) viene usata sulla sinistra dei simboli # e fa sì che gli spazi iniziali, che risultano da una giustificazione a destra di un valore, vengano riempiti con asterischi; quindi:

```
? USING "**#.#";4.2 dà come risultato **4.2
```

```
? USING "**##.#";4.2 dà invece ***4.2
```

Il segno di doppia lira (che qui indichiamo con $$) provoca la stampa del simbolo di lira prima del valore numerico in questione. Questo carattere di controllo non può essere usato in coincidenza del "carat" (^^^^).
Se si pone una virgola alla sinistra del punto decimale, il numero viene stampato con una virgola a sinistra di ogni gruppo di tre cifre, vale a dire:

```
? USING "####,.#";2222.2 provoca la stampa di 2,2222.2
```

Infine il quadruplo carattere (^) viene usato per stampare un valore numerico in forma esponenziale; come esempio si consideri:

```
? USING :##.#^^^^";22.4 implica la stampa di 2.2E+01
```

Lo schermo può essere pulito in tutte le modalità di visualizzazione mediante il comando CLS. In entrambe le modalità testo, un'alternativa è la stampa di CHR$(12). La coordinata X da POS(x) (dove X è un argomento fittizio, che può quindi essere sostituito da un qualsiasi valore).

MANIPOLAZIONE DELLA RAM VIDEO

Per ottenere che i caratteri siano visualizzati con colori diversi da quelli impostati globalmente per mezzo del comando COLOR nella modalità testo a 32 colonne, si deve andare a modificare il byte appropriato nella tabella dei colori della VRAM, scegliendo una nuova combinazione primo-piano/sfondo.
Come già accennato, ciascuno dei 32 byte della tabella dei colori determina i colori di un insieme di 8 dei 256 caratteri disponibili. Come conseguenza di ciò, per visualizzare del testo in colori diversi da quelli di default è necessario copiare parte dell'insieme dei caratteri in un'altra sezione della tabella di generazione dei modelli.
Ecco come è allocata la memoria video nella modalità testo a 32 colonne:

TABELLA GENERATRICE DEI MODELLI: 0 - 2047
TABELLA DEI COLORI: 8192 - 8223
TABELLA DEGLI ATTRIBUTI DEGLI SPRITE: 6912 - 7039
TABELLA DEI NOMI: 6144 - 6911
TABELLA DEI MODELLI DEGLI SPRITE: 14336 - 16383

Si noti che i caratteri grafici occupano i primi 2 blocchi di definizione nella tabella dei modelli. Questo perché i primi 32 codici carattere non sono stampabili. I caratteri grafici possono quindi essere visualizzati sullo schermo solo stampando (PRINT) CHR$(1) + CHR$(65).
Un altro carattere non standard ha codice 255: provoca la visualizzazione inversa del carattere su cui è posizionato il cursore.
Viene aggiornato circa ogni 1/50esimo di secondo.

ESEMPIO DI TESTO MULTICOLORE

Per ottenere che un testo venga visualizzato in lettere maiuscole con primo piano giallo e sfondo nero, sono necessari i seguenti passi:
Copiare le 26 definizioni dei caratteri in un'altra sezione della tabella di generazione dei modelli. L'insieme di caratteri viene definito usando 8 byte per ciascun carattere. L'insieme di caratteri viene definito usando 8 byte per ciascun carattere.
Nella modalità testo a 32 colonne la tabella dei modelli si estende, nella VRAM, dalla locazione 0 fino alla 2048.
Poiché ogni carattere necessita di otto byte di definizione, il primo byte che deve essere copiato si trova alla locazione:

```
ASC("A") * 8 = 520
```

e il byte finale della definizione dei caratteri sarà

```
ASC("Z")*8+7=727
```

I 'nuovi' caratteri maiuscoli sovrascriveranno parte dell'insieme di caratteri esistenti. Sostituiremol i caratteri 145-170, che sono definiti da 145*8=1160 a 170*8+7-1467, come segue:

```
10 SCREEN 1:FOR X=0 TO 207 : VPOKE 1160 + X , VPEEK (520 + X) : NEXT X
```

I quattro byte della tabella dei colori che determinano i colori che assumeranno i 32 caratteri dal 144 al 176 risiedono alle locazioni 8210-3.
Il valore contenuto dai quattro bit più significativi in ciascuno di questi determina il colore del primo piano, mentre quelli meno significativi influiscono sul colore dello sfondo.
Quindi, con

```
VALORE = GIALLO * 16 + NERO = 10 * 16 + 1 = 161
```

DENTRO L'MSX (Parte IVª)

la prossima linea di programma è:

```
20 FOR X=0 TO 3:VPOKE X+8210,161:NEXT X
```

Se ora si prova a far visualizzare il CHR$(145) apparirà una 'A' gialla in capo nero.
Si noti che se si usa il comando COLOR tutti i 32 byte della tabella del colore sarebbero impostati con i colori prescelti.
Per ottenere nuovamente dei colori alternativi sarebbe necessario ripetere la linea 20.
I caratteri sono giustificati a sinistra: quindi quando vengono visualizzati su uno sfondo che contrasta con il colore dello schermo, può succedere che la parte sinistra del carattere appaia confusa.
Questo metodo per determinare i colori dello schermo, sebbene macchinoso per certi aspetti, permette di realizzare delle schermate efficaci dal punto di vista grafico in modo piuttosto semplice.
Per esempio, una schermata che consista principalmente di 4-8 caratteri può essere 'animata' o resa lampeggiante manipolando solo due byte della tabella del colore.
Il programma che viene presentato di seguito consente di ridefinire interamente l'insieme dei caratteri, di salvarlo e caricarlo da nastro.
I tasti di controllo del cursore sono usati per muoversi sulla matrice dei caratteri e la barra di spaziatura viene utilizzata per invertire il valore del particolare bit su cui si è posizionati.
Eccovi, per concludere, un'esempio di programma per la definizione dell'insieme di caratteri:

```
100 'DEFINIZIONE CARATTERI'
110 FOR X=0 TO 12:READ A$,B$:POKE 380000+X,VAL("&H"+A$):POKE 38200+X,VAL("&H"+B$):NEXT X
120 DEFUSR=38000:DEFUSR2=38200
130 DATA 21,21,00,40,00,9C,11,11,40,00,9C,00,01,00,00,08,08,CD,CD,59,5C,00,00,C9,C9
140 ON KEY GOSUB 240,270,290,430:KEY(1) ON:KEY(2) ON:KEY(3) ON:KEY(4) ON
150 ON STRIG GOSUB 400:STRIG(0) ON
160 KEY OFF:COLOR 14,1,1:SCREEN 1,0:CLS
170 LOCATE 2,1:PRINT "F1..Save F3..Change CHR$:LOCATE 2,3:PRINT "F2..LOAD F4..COPY CHR$"
180 FOR X=0 TO 7:VPOKE 14336+X,VPEEK(224+X):NEXT X:PUT SPRITE 0,(144,47),8,0
190 FOR X=373 TO 375:VPOKE X,0:NEXT X:VPOKE 371,24:VPOKE 372,24:
200 A$=STRING$(8,46):FOR X=0 TO 7:LOCATE 16,X+6:PRINT A$:NEXT X
210 GOSUB 290:X1=1:Y1=1
220 X2=STICK(0):IF X2=0 THEN 220
230 ON X2 GOTO 350,220,360,220,370,220,380
240 Z=UST(2):LOCATE 2,20:PRINT "REGISTRA E POI RITORNA"
250 A$=INPUT$(1):IF ASC(A$)‹›13 THEN 250 ELSE BSAVE "CHARS",40000,42047
260 LOCATE 2,20:PRINT STRING$(28,32):RETURN
270 LOCATE 2,20:PRINT "LOADING...":BLOAD"CHARS":Z=USR2(2)
280 LOCATE 2,20:PRINT STRING$(28,32):RETURN
290 LOCATE 0,20:PRINT "DIGITA N. DI CHR$‹Invio›":GOSUB 470:CN=XX
300 FOR X=0 TO 7:A$(X)=BIN$(VPEEK(CN*8+X)):NEXT X 310 FOR X=0 TO 7:IF LEN(A$(X))‹8 THEN A$(X)=STRING$(8-LEN(A$(X)),79)+A$(X)
320 FOR BT=1 TO 8:LOCATE 15+BT,X+6:IF MID$(A$(X),BT,1)="1" THEN PRINT CHR$(219) ELSE PRINT CHR$(46)
330 NEXT:NEXT:LOCATE 2,11:PRINT "CARATTERE ";CN:LOCATE 7,13:PRINT CHR$(CN)
340 RETURN
350 IF Y1=1 THEN 220 ELSE Y1=Y1-1:GOTO 390
360 IF X1=8 THEN 220 ELSE X1=X1+1:GOTO 390
370 IF Y1=8 THEN 220 ELSE Y1=Y1+1:GOTO 390
380 IF X1=1 THEN 220 ELSE X1=X1-1
390 PUT SPRITE 0,(136+X1*8,39+Y1*8),8,0:FOR X=1 TO 80:NEXT:GOTO 220
400 LOCATE 15+x1,5+Y1:IF MID$(A$(Y1-1),X1,1)="1" THEN MID$(A$(Y1-1),X1,1)="0":PRINT CHR$(46) ELSE MID$(A$(Y1-1,X1,1)="1":PRINT CHR$(219)
410 NN=CN*8+Y1-1:IF MID$(A$(Y1-1),X1,1)="1" THEN VPOKE NN,VPEEK(NN)ORE(2^(8-X1)) ELSE VPOKE NN,VPEEK(NN)AND(255-(2^(8-X1)))
420 RETURN
430 LOCATE 0,20:PRINT "CHR$ DA COPIARE‹Invio›":GOSUB 470:C1=XX
440 LOCATE 0,20lORINT "CHR$ DA RIMPAZZARE":GOSUB 470:C2=XX
460 LOCATE 0,20:PRINT STRING$(28,32):RETURN
470 C1$=""
480 X$=INKEY$:IF X$="" THEN 480
490 IF ASC(X$)‹32 AND ASC(X$)›27 THEN 480
500 IF ASC(X$)‹›13 THEN C1$=C1$+X$:GOTO 480 ELSE XX=VAL(C1$)
510 IF XX›255 OR XX‹0 THEN 470
520 LOCATE 0,20:PRINT STRING$(28,32):RETURN
```

Eccoci giunti al nuovo appuntamento con il nostro corso di programmazione.

Parleremo della

Programmazione o Notazione Lineare Strutturata

che permette di generalizzare qualsiasi algoritmo ponendosi alla base di qualsiasi linguaggio.

PROGRAMMAZIONE LINEARE STRUTTURATA

Trent'anni fa i Diagrammi a Blocchi (DaB) erano l'unico strumento usato per esprimere un algoritmo.

Per farlo poi eseguire da un calcolatore, si procedeva alla traduzione in un linguaggio di programmazione (Cobol o Fortran).

Vent'anni fa circa sono finalmente sorti degli strumenti alternativi alla "diagrammazione" cioè la programmazione mediante diagrammi a blocchi.

Questi strumenti hanno contribuito ad evidenziare alcuni difetti di questa forma espressiva, soprattutto sotto la spinta di una nuova generazione di linguaggi, che trovano nel Pascal il loro rappresentante di maggior successo.

I maggiori difetti riscontrati nei diagrammi di flusso o a blocchi sono:

A) Molto spesso sono illeggibili, soprattutto quando le loro dimensioni superano quelle di un semplice esercizio didattico.
La lettura andrebbe fatta un po' dall'alto al basso, un po' dal basso all'alto senza un ordine preciso.

B) Sono facilmente esposti ad errori logici : bastano pochi accavallamenti di cicli per perdere il filo del controllo, e quindi, della risoluzione del problema.

A tutto ciò si aggiungono poi i seguenti inconvenienti:

C) Scarsa praticità dovuta alla natura grafica bidimensionale poiché tutti i linguaggi di programmazione esistenti sono unidimensionali.
Sul piano concreto, tutti gli esecutori automatici comprendono un linguaggio fatto di stringhe, cioè sequenze di caratteri.
Più precisamente, tutto il programma viene visto come un'unica stringa di caratteri divisa in linee poste una di seguito all'altra.
Volendo far eseguire automaticamente un diagramma a blocchi, saremmo perciò costretti a compiere una traduzione in un linguaggio di programmazione disponibile, di natura unidimensionale, spesso modificando la struttura ordinaria.

D) Difficoltà di riconoscimento della struttura di controllo.
I blocchi possono essere disposti con la massima libertà sul foglio di carta, senza alcuna regola particolare.
Può così accadere di non riconoscere due diagrammi che differiscono solo per la dislocazione dei blocchi, ovvero ci potrebbero essere due rappresentazioni diverse dello stesso algoritmo.

Teniamo comunque sempre a mente che tutti gli inconvenienti sottolineati per i DaB, hanno come unica radice l'uso indisciplinato delle frecce di controllo.

Per ovviar e a tutti questi problemi, è stata creata la

PROGRAMMAZIONE LINEARE STRUTTURATA.

Vediamo ora alcuni degli schemi di composizione detti anche costrutti di controllo strutturati.

Ognuno di essi verrà descritto sintatticamente da alcune regole che sono sostanzialmente le stesse che si ritrovano nei linguaggi di programmazione come il Pascal o il C.

COSTRUTTO SEQUENZA O BLOCCO BEGIN-END

Questo schema prende ordinatamente un numero arbitrario di blocchi strutturati B1, B2, ...,Bn, e ne costruisce la sequenza seguente:

Sintassi PLS

```
Begin
B1
.
B2
.
Bn-1
.
Bn
End*8
```

E' evidente che si impone di eseguire prima B1, poi B2, fino a Bn.
Se qualcuno dei blocchi si ferma, tutto il costrutto si ferma.

COSTRUTTO IF-THEN-ELSE
(se-allora-altrimenti)

Questo schema valuta inizialmente una condizione ‹cond›; se questa è vera, viene successivamente eseguito un blocco strutturato B1, altrimenti viene eseguito un blocco strutturato B2 con sintassi PLS:

```
if ‹cond› then B1 else B2
```

COSTRUTTO WHILE
(finché la condizione è vera ripeti)

Questo schema prevede una condizione ‹cond› da valutare inizialmente.

Se questa è falsa si esce immediatamente dal costrutto, altrimenti si esegue ciclicamente un blocco strutturato B, detto corpo, uscendo se e quando la condizione ‹ cond› diventerà falsa.

Sintassi PLS:

```
while ‹cond› do B
```

Vediamo ora un esempio che utilizza questi tre costrutti con un programmino in programmazione lineare strutturata che calcola il minimo comune multiplo fra due numeri naturali A e B.

```
begin
leggi A, B.
MA = A;
MB = B;
While MA ‹› MB do if MA › MB then MA = MA+A
else MB = MB+B.

MCM = MA;
scrivi MCM
end
```

E' importante notare che si esce dal ciclo while quando la condizione è falsa, nel nostro caso quando MA=MB.

COSTRUTTO CASE
(selezione a molte vie)

Abbiamo visto che il costrutto if-then-else consente di distinguere fra due possibilità corrispondenti alla verità o falsità di una certa condizione.

In pratica vi sono molti casi in cui si vorrebbe poter selezionare contemporaneamente una fra un numero più elevato di possibilità.

Pensiamo ad esempio ad un problema di questo tipo:
leggere un numero fra 1 e 7 e scrivere in uscita il nome del giorno corrispondente :
Lunedì, Martedì, Mercoledì, Giovedì, Venerdì, Sabato, Domenica.

Non vorremmo fare tutta la serie di confronti del tipo:

```
if I=1 then scrivi lunedì;
if I=2 then scrivi martedì:
.
.
.
if I=7 then scrivi domenica;
```

Quindi scriveremo secondo la sintassi della PLS :

```
leggi N
case N of
1: scrivi LUNEDI';
2: scrivi MARTEDI';
.
.
.
7: scrivi DOMENICA;
otherwise scrivi 'non accetto'
end-case
```

COSTRUTTO REPEAT-UNTIL
(ripeti fino a quando)

Si tratta di un tipo di ciclo in cui si ripete una sequenza di blocchi fino al verificarsi di una certa condizione.

In sintassi PLS:

```
repeat
‹B1›;
‹B2›;
.
.
.
‹BN›;
until ‹Condizione›.
```

COMPUTER & PROGRAMMAZIONE (Parte IVª)

Occorre evidenziare subito due importanti differenze fra cicli while e cicli repeat until:

nei cicli while il test di terminazione si fa all'inizio delle iterazioni e quindi l'esecutore potrebbe anche non effettuare alcun ciclo;

nei cicli repeat-until la condizione viene testata alla fine e il corpo del ciclo viene eseguito almeno una volta.

nei cicli while si esce quando la condizione è falsa mentre in quelli repeat- until quando la condizione è vera.

Supponiamo di voler scrivere un semplice algoritmo che scrive il numero di cifre decimali di cui è composto un numero N dato in ingresso:

```
begin
leggi N;
NC=0;
repeat NC=NC+1;N=N/10 until N=0;
scrivi NC;
end
```

COSTRUTTO FOR
(iterazioni enumerate)

Nella pratica programmativa, accade molto spesso di voler eseguire un numero prefissato di iterazioni.

Se disponiamo solo dei cicli while e repeat-until saremmo costretti ad introdurre delle variabili per conteggiare il numero delle iterazioni.

Se a ciò si aggiunge il fatto che questa variabile contatore viene utilizzata nello stesso ciclo (si pensi, ad esempio, alla somma dei primi N numeri) si comprende l'importanza del costrutto for che è esplicitamente destinato a questo scopo.

Purtroppo, a parte l'idea generale, non esiste un grande accordo su cosa debba fare esattamente il costrutto for.

Forniamo qui la versione che ci sembra più semplice ed opportuna in sintassi PLS:

```
For indice = inizio to fine do ‹B›
```

Occorre definire che indice è una variabile che fa le veci del contatore e che viene posta a zero all'inizio del ciclo, al valore dell'espressione inizio e incrementata dopo ogni esecuzione del corpo ‹B› del ciclo.

L'uscita è condizionata al superamento di un certo valore contenuto in una variabile speciale LIM, associata al ciclo, che viene caricata all'ingresso con il valore dell'espressione ‹fine› senza che possa subire assegnamenti nel corso dell'esecuzione dei cicli.

Supponiamo di voler scrivere la tabella dei fattoriali dei numeri da 1 a N :

```
begin
leggi N;
FATT=1;
FOR I=1 TO N do
begin
FATT=FATT*I;
scrivi "IL FATTORIALE DI",I,"E'",FATT
end;
end
```

Facciamo presente che il test di terminazione viene eseguito all'ingresso del ciclo.

Così il numero dei cicli è determinato dalla seguente regola : se ‹fine› è maggiore di ‹ inizio› allora vengono eseguiti zero cicli, altrimenti vengono eseguiti esattamente

‹fine›-‹inizio›+ 1 cicli.

Naturalmente si fa l'ipotesi che la variabile indice non venga modificata nel corso dell'esecuzione dei cicli.

In effetti, tutti i linguaggi di programmazione proibiscono ogni tentativo di questo tipo.

Una buona regola di programmazione stabilisce di non fidarsi del valore dell'indice all'uscita dal ciclo a causa delle diverse interpretazioni del FOR nei i vari linguaggi.

Facciamo infine presente che molti linguaggi prevedono la possibilità di specificare un incremento della variabile indice diverso da quello unitario e anche negativo.

(continua)

TENDENZE

E' terminato un altro anno, il primo di un nuovo decennio, e con la fine di un anno e l'inizio di quello nuovo si fa sempre un bilancio di ciò che è stato.

Dal bilancio dell'ultimo anno e dell'ultimo decennio possiamo cercare di capire ciò che sarà, dove andrà il mercato e l'intero mondo dell'informatica.

Vediamo quali sono le tendeze del mercato per questo nuovo anno, per il prossimo decennio.

TELECOMUNICAZIONI

Le telecomunicazioni contribuiscono per oltre metà del business complessivo dell'information technology ed il settore tecnologico in maggiore sviluppo e che sta conoscendo le più profonde trasformazioni.

Il 1989 ha visto finalmente la concretizzazione di un dibattito che aveva gettato le sue basi già vent'anni fa. Stiamo parlando della

Rete Numerica Integrata "ISDN".

A partire dal 1988 infatti hanno avuto il via le prime realizzazioni concrete e i relativi apparati stanno diventando finalmente disponibili sul mercato.

Nel corso dell'anno passato l'introduzione delle reti ISDN nei paesi Cee hanno proceduto in modo abbastanza spedito, ed è stato completato circa il 60/70% dei lavori di standardizzazione.

A fine anno il servizio commerciale ISDN era offerto in quattro paesi europei e in altri quattro erano disponibili servizi su basi sperimentali.

Tutti gli altri paesi Cee prevedono l'introduzione del nuovo sistema entro il 1992.

Anche le tecnologie di trasmissione hanno riscontrato un evoluzione degna di nota.

Nelle moderne reti di trasmissione si assiste ormai a una estensione del ruolo dei sistemi trasmissivi dal puro trasporto ad altre aree in forte sviluppo come la radiotelefonia mobile, le reti di distribuzione e la gestione degli istradamenti.

E in questo senso le aziende che operano nel settore, vista questa diversificazione, hanno maggiori possibilità di ampliare le proprie prospettive.

A fronte di questo aspetto l'introduzione delle fibre ottiche sta creando la condizione per assicurare la connettività globale e la capacità tecnologica di supportare applicazioni integrate.

Ma il 1990 è stato sicuramente l'anno della telefonia mobile.

La diffusione delle reti cellulari ha favorito lo sviluppo di una serie di servizi a partire dall'ormai tradizionale telefono in macchina, ai Telepoint, ai telefoni installati sugli aerei, ma soprattutto alla telefonia personale,

il Personal Communications Networks.

Nel nostro paese, che proprio sulla telefonia personale sta attraversando un momento di vera "corsa al telefono portatile", la scelta tecnologica è caduta su un sistema-ponte analogico che opera sulla frequenza dei 900 Mhz.

Un ultimo sguardo a un altro fenomeno tecnologico, ma ormai anche di costrunme, che nel 1990 ha compiuto un vero e proprio rilancio.

Stiamo parlando naturalmente del

Videotel

che iniziati i suoi primi passi nel lontano 1986, e dopo una fase di sperimentazione contrassegnata da una serie di difficoltà, sta finalmente decollando.

Il concessionario pubblico ha seguito la felice esperienza del Minitel francese e ha avviato una rivoluzione tecnologica e di marketing che ha consentito da una parte una vertiginosa crescita quantitativa e qualitativa dei servizi e dei fornitori di informazioni (in Francia sono chiamati "Serveurs") e dall'altra un'impennata dei possessori dei terminali, visto che vengono noleggiati a un canone mensile di L. 7000.

E a fronte di questa "rivoluzione" i risultati non hanno tardato ad arrivare: alla fine del 1989 gli abbonati erano pressochè raddoppiati, raggiungendo la ragguardevole cifra di 77.500.

E gli abbonati sono in continua crescita, mese dopo mese.

IL SETTORE INFORMATICO

Il biennio 1989-90 è stato protagonista di una sequenza di eventi che consentono di prevedere con sufficiente approssimazione le tendenze dei singoli mercati nel corso dell'immediato futuro, e questo a dispetto delle notizie piuttosto negative che hanno scosso il mercato mondiale soprattutto nel comparto dell'informatica.

Si è molto parlato, tra la fine del 1989 e l'inizio del 1990 della crisi dell'industria informatica mondiale, che si è esplicitata in una forte riduzione di utili per alcune delle maggiori aziende americane ed europee, tanto da spingere molti osservatori a dichiarare chiuso un ciclo di crescita a tasso elevato e ad annunciare una fase di profonda ristrutturazione.

In effetti l'evoluzione dell'informatica mondiale non sembra attraversare una semplice crisi congiunturale: le tendenze in atto sono frutto di una trasformazione che a breve termine sta producendo problemi di tipo circoscritto e apparentemente negativi, ma a medio termine potrebbe portare a nuove opportunità di sviluppo del mercato.

A questo proposito è interessante sottolineare che uno dei problemi principali che ha determinato la congiuntura sfavorevole e senz'altro il forte disallineamento tra l'elevata velocità di crescita della domanda, determinata dall'incapacità dell'utenza di tradurre in applicazioni le innovazioni tecnologiche lanciate sul mercato dai costruttori.

Ed è proprio su questo punto che oggi si registrano i maggiori risultati sfavorevoli, ma che in futuro si potrà ricostruire la capacità di fare profitto dei produttori di tecnologia informatica: se questi ultimi infatti sapranno fornire agli utenti gli strumenti culturali per sfruttare la tecnologia di volta in volta disponibile, la capacità di assorbimento del mercato risulterà aumentata, e il mercato si ristabilizzerà sui livelli di crescita a cui ci si è abituati per un intero decennio.

Il personal computer tuttavia non ha risentito che in minima parte della congiuntura, e soprattutto in Italia i dati di vendita sono rimasti su livelli soddisfacenti.

Il segmento di mercato che risulta ancora trainante è quello dei pc di classe 286, cioè basati su microprocessore Intel 80286, che hanno un parco installato di oltre 800mila unità, quasi il 35% del totale.

I sistemi di fascia bassa, basati su microprocessore 8086, che secondo le previsioni degli anni passati avrebbero dovuto registrare un declino, si sono mantenuti su buoni livelli di vendita e diffusione, e questo trend verrà conservato ancora per almeno due anni. Il parco installato in Italia è ormai di 813mila macchine, quasi il 50% del totale.

I personal computer basati su microprocessore 80386sx, presentati alla fine del 1988, hanno conseguito un notevole successo: 54 mila unità vendute, grazie al rapporto fra prezzo e potenza di calcolo che li ha suggeriti come soluzione ottimale per utenti di sistemi della fascia inferiore, ma anche soluzione meno costosa per utenti che in precedenza avrebbero optato per sistemi 80386.

Questi ultimi hanno confermato nel corso del 1990 il successo atteso, ponendosi in aperta concorrenza con i microcomputer di fascia bassa e le workstation tecnico-scientifiche grazie alla disponibilità di schede acceleratrici, monitor ad alta risoluzione, hardware e software di rete locale e memorie di massa di sempre maggiori dimensioni.

Un discorso a parte merita infine il segmento dei personal computer portatili, che ha imboccato la strada del successo duraturo e progressivo grazie alla loro capacità di racchiudere in un ingombro limitato le caratteristiche e la potenza di calcolo dei sistemi da tavolo.

Mentre in passato i costruttori presentavano in chiave di marketing il plus della trasportabilità e dell'autonomia dalla rete elettrica, oggi l'approccio è mutato a favore della compattezza e del minimo ingombro sulla scrivania, cercando con questo di catturare l'attenzione di professionisti, imprenditori e dirigenti che cercano nel personal computer qualcosa in più rispetto al pc da scrivania.